

101 }

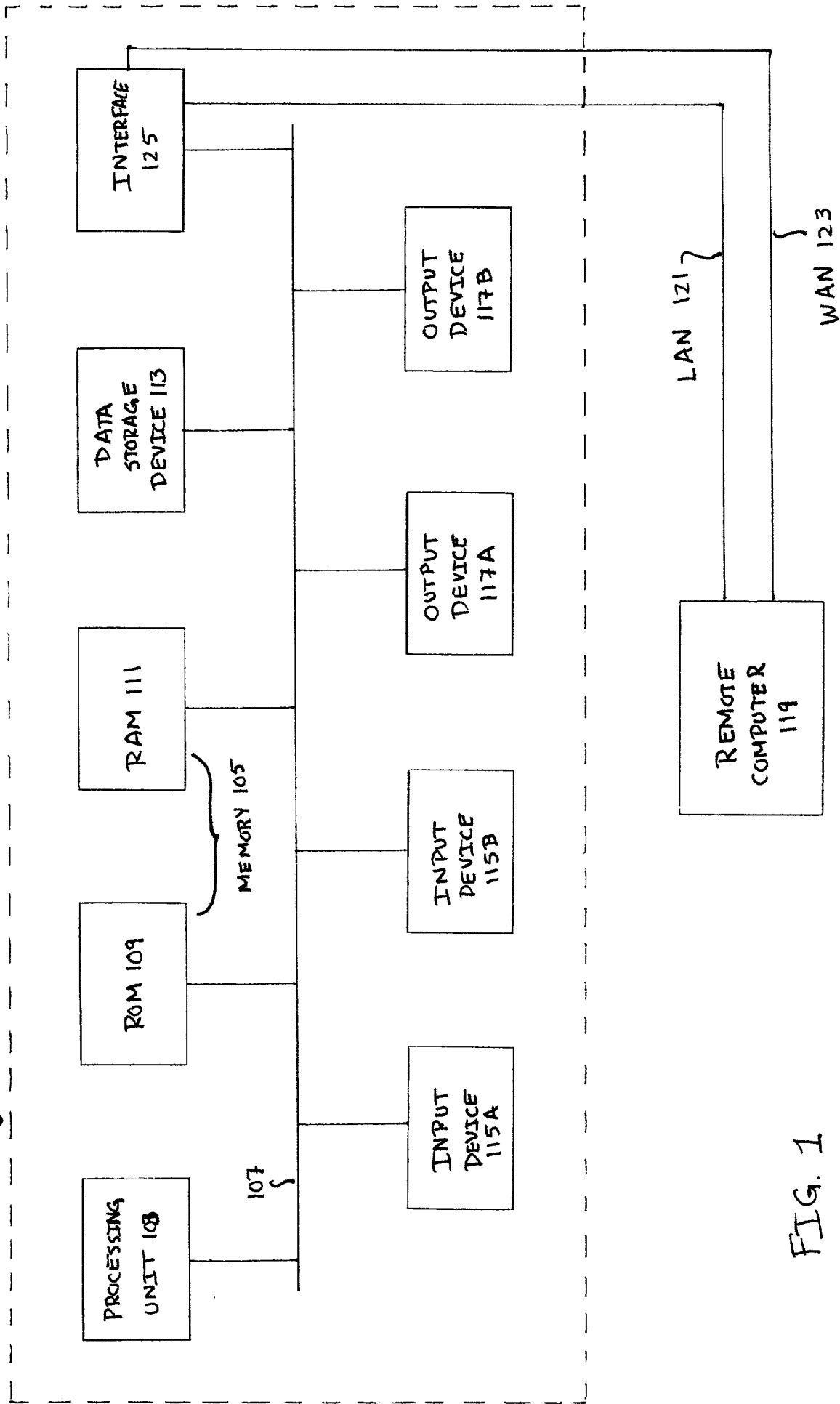


FIG. 1

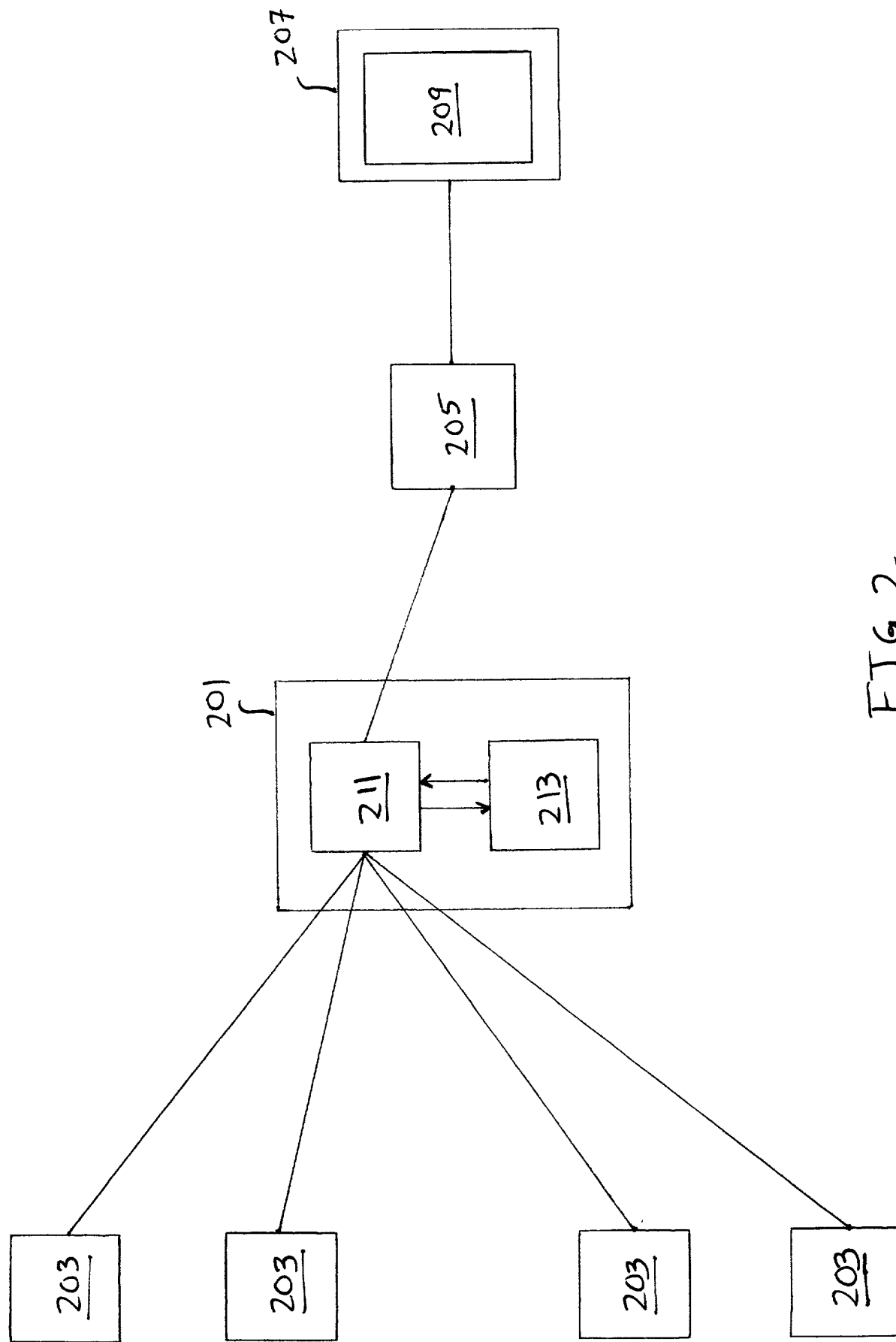
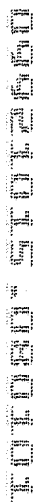
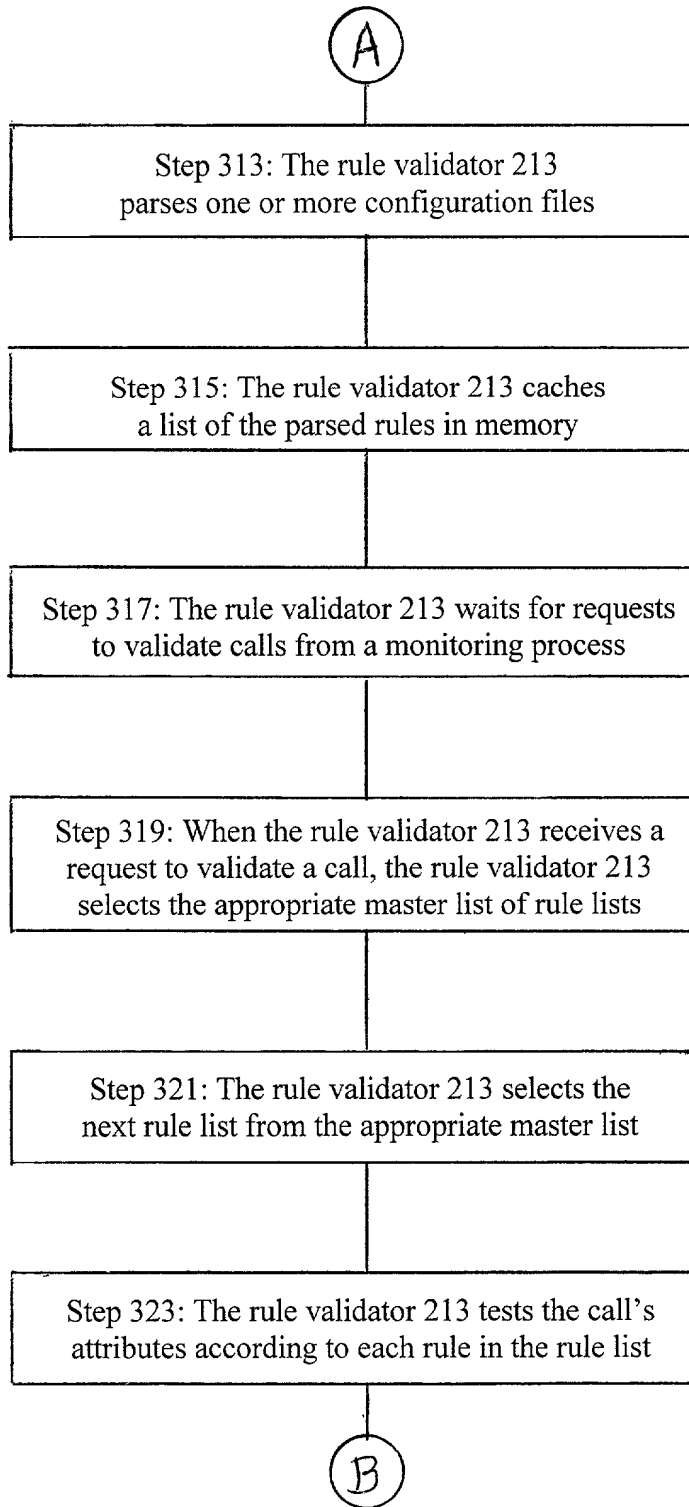


FIG. 2

[illegible][illegible][illegible][illegible]



**FIG. 3B**

```
graph TD; Start((B)) --> Step325{Step 325: The rule validator 213 determines if there are any rule lists in the master list that have not been used to test the call}; Step325 -- N --> Step329[Step 329: The process monitor determines if the rule validator 213 has passed the call]; Step325 -- Y --> Step327[Step 327: The rule validator 213 returns the test result to the process monitor]; Step327 --> Step329; Step329 --> Step331[Step 331: The process monitor forwards the call to the directory access protocol server device 205]; Step331 --> Step333[Step 333: The process monitor provides the appropriate error messages to the client computer 203 that originated the call];
```

The flowchart illustrates the call processing method, starting with a connector 'B' leading to a decision diamond (Step 325). The diamond contains the text: 'Step 325: The rule validator 213 determines if there are any rule lists in the master list that have not been used to test the call'. From the diamond, two paths emerge: a 'Y' path leading to a process rectangle (Step 327) and an 'N' path leading to another process rectangle (Step 329). Step 327 contains the text: 'Step 327: The rule validator 213 returns the test result to the process monitor'. Step 329 contains the text: 'Step 329: The process monitor determines if the rule validator 213 has passed the call'. Both Step 327 and Step 329 lead to Step 331, which contains the text: 'Step 331: The process monitor forwards the call to the directory access protocol server device 205'. Step 331 leads to the final step, Step 333, which contains the text: 'Step 333: The process monitor provides the appropriate error messages to the client computer 203 that originated the call'.

